第6章　软件架构与敏捷开发

王璐璐 wanglulu@seu.edu.cn
廖力　lliao@seu.edu.cn

目录

6.1软件开发的发展简史

- 软件开发的历史，至今其发展过程经历了四个明显的阶段
  - 松散的软件开发阶段
  - 基于软件工程思想的开发阶段
  - 基于敏捷过程的软件开发阶段
  - 智能化软件开发阶段

6.2 敏捷开发

- 敏捷开发的基本理念
  - （1）强调个体和互动比强调过程和工具更好（Individuals and interactions over processes and tools）；
  - （2）强调获得可运行的软件比强调完成详尽的文档好（Working software over comprehensive documentation）；
  - （3）强调与客户合作比强调进行详细的合同谈判好（Customer collaboration over contract negotiation）；
  - （4）强调响应变化比强调遵循既定的计划好（Responding to change over following a plan）

6.2 敏捷开发

- **具有代表性的敏捷开发的方法**
  - 极限编程
  - Scrum方法
  - 特征驱动软件开发
  - Crystal方法
  - 精益开发方法
  - 动态系统开发方法
  - 自适应软件开发方法
  - 等等

6.2 敏捷开发

- 敏捷开发在实践中表现为一种迭代、增量和持续集成的开发方法。
  - 迭代反映了项目的开发节奏，是一个多周期的开发过程。
  - 增量说明了项目的实际进展，整个项目就是由很多增量构成的。
  - 持续集成反映了集成增量的过程是持续进行的。

6.2 敏捷开发

- 1. 软件架构与敏捷开发的出发点是一致的。
  - 软件架构与敏捷开发都是一个权衡的过程：软件架构设计需要权衡涉众们的各种需求，在众多的解决方案中确定唯一的架构设计方案，从而保证软件开发的有效性。敏捷开发是在软件开发过程混沌和大量开发管理活动加入的两个极端中做出的一种权衡。
  - 软件架构与敏捷开发目的都是为了提高软件开发效率、提高软件质量、降低软件成本，将开发团队的价值最大化。

6.2 敏捷开发

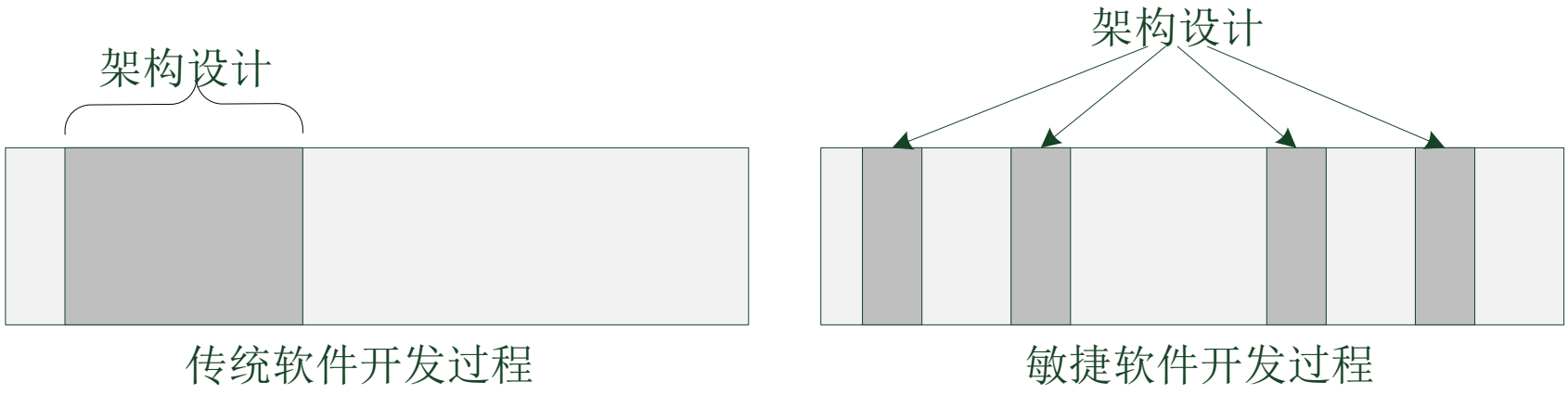
- 2.敏捷开发也需要重视软件架构。
  - 在敏捷开发的支持者看来，传统的软件架构设计与敏捷思想相违，因为过多的预先设计使得软件开发过程在面对变化时缺乏灵活性，其管理成本极有可能由于后续的重构而成为无用功。
  - 根据学术界与企业界的研究和调查，发现软件架构设计对于敏捷开发来说也是必要的。两者在软件开发实践中能够共同存在，且互相促进。

6.2 敏捷开发

- 3. 敏捷开发改变了软件架构的设计方式。
  - 敏捷开发非常重视软件的架构设计，但是轻架构的详细设计。
  - 敏捷思想中将传统的架构设计分成：种子架构设计+详细架构设计。
  - 种子架构设计关注软件系统的骨架或轮廓的设计。
  - 敏捷开发将详细架构设计转移到Code编码阶段、重构阶段、单元测试阶段等。
  - 分离后，敏捷软件种子架构的内容包括：软件的架构层次，重要模块、重要类的说明（无须设计全部的类和方法）等。

6.2 敏捷开发

- 3.敏捷开发改变了软件架构的设计方式。
  - 传统软件开发和敏捷软件开发在架构设计方式上的区别如图：

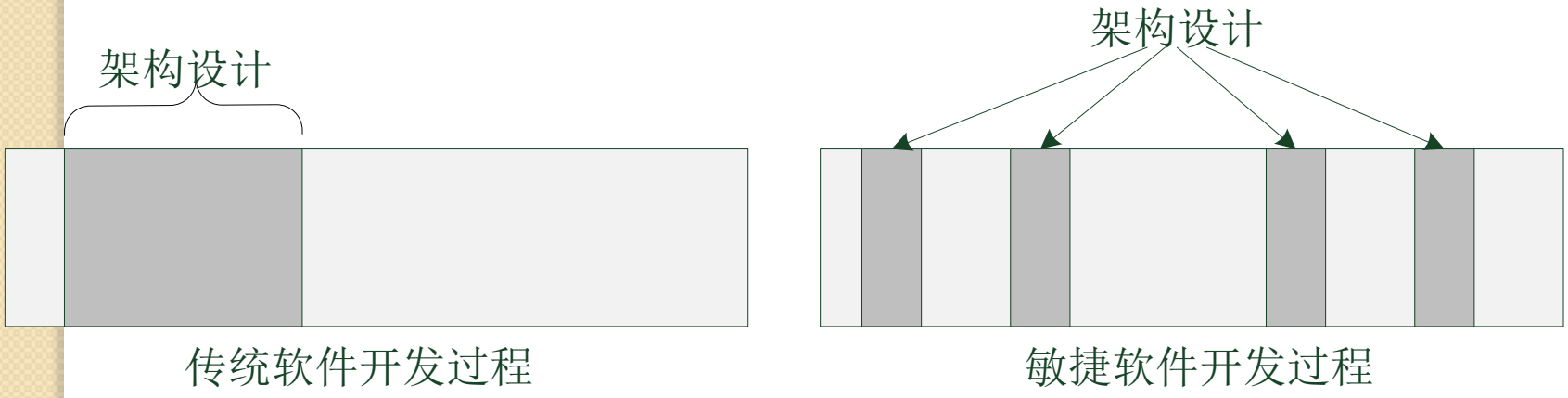


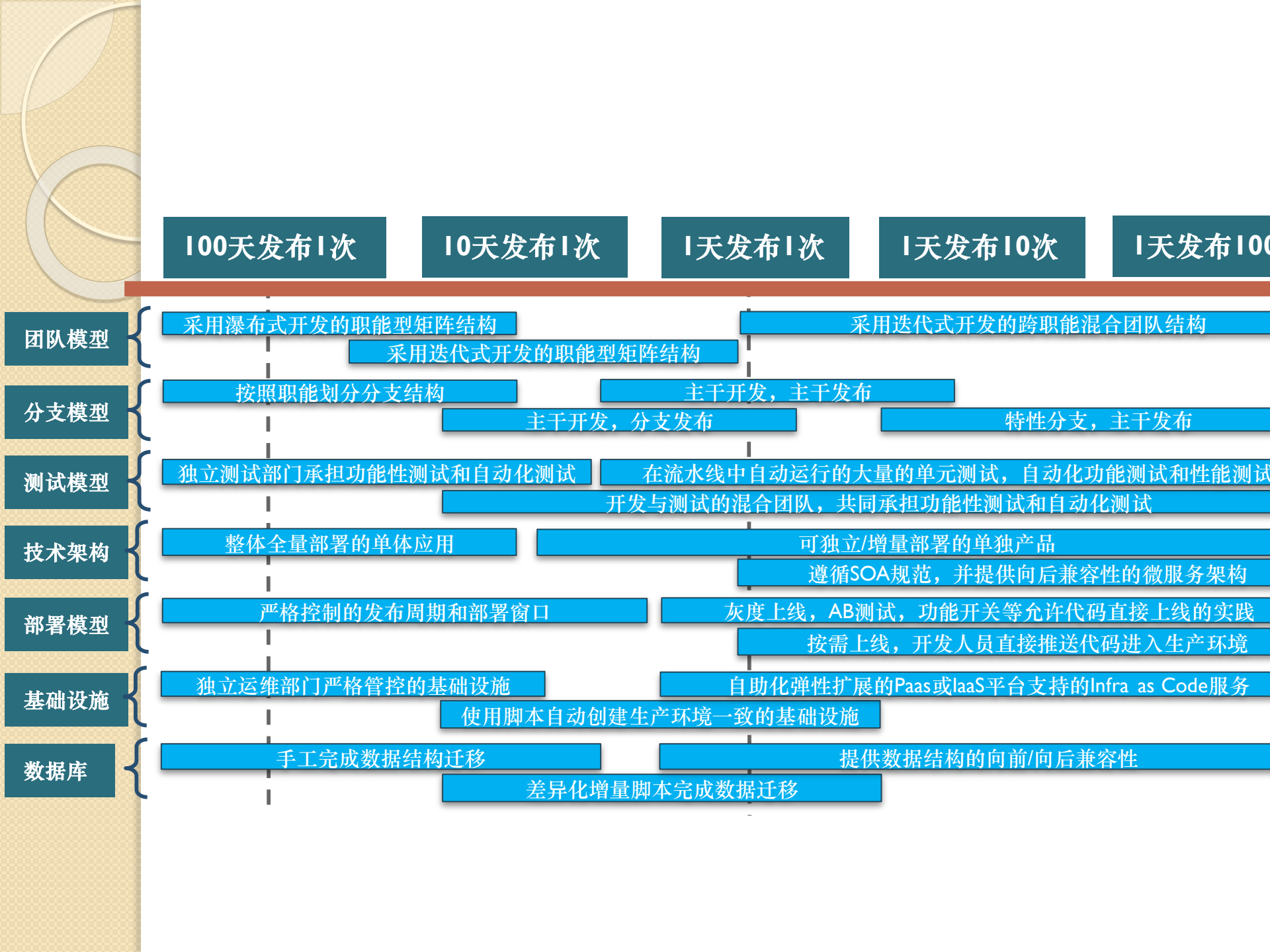
6.2 敏捷开发

- 3.敏捷开发改变了软件架构的设计方式。
  - 敏捷开发中架构设计的原则
    - 重要、关键的设计决策必须在软件开发前确定，对于其他详细的设计决策，则需要时再做。
    - 开发过程中，所有的利益相关者都必须全程在一起，便于充分的交流和权衡。
    - 将需求文档化，重视需求，明确表示出不同需求之间的权衡过程。
    - 开发人员要及时的验证架构。
    - 如果必须要进行修改，改动越早越好。
    - 不要突发奇想地修改架构，修改架构需要大家一起深思熟虑。

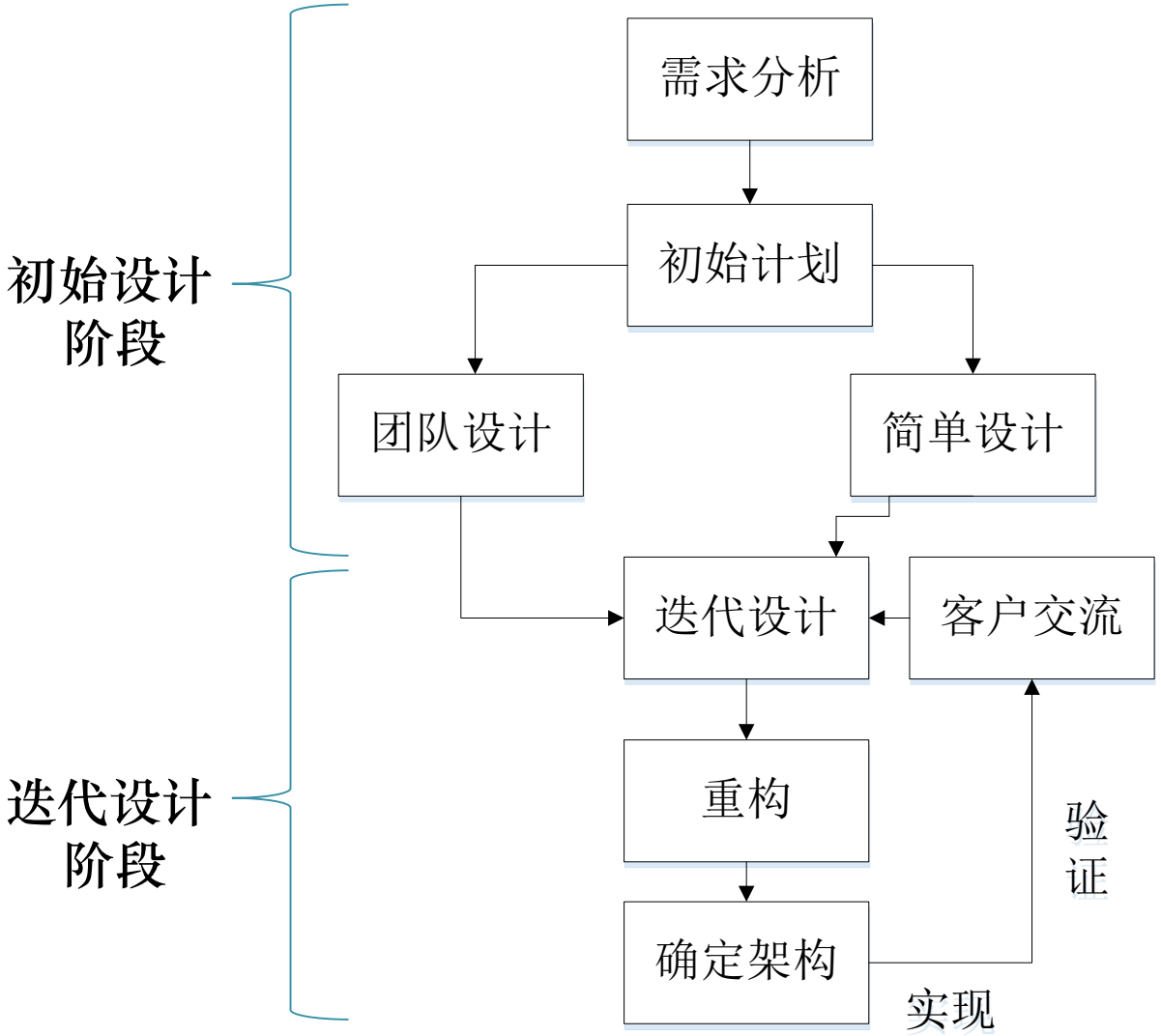
6.3敏捷开发过程中的软件架构设计

- 敏捷开发把传统软件开发前期的详细架构设计，分散到了整个敏捷开发软件过程中，以达到提高效率、减少风险的目的。




100天发布1次
10天发布1次
1天发布1次
1天发布10次
1天发布100
团队模型
采用瀑布式开发的职能型矩阵结构
采用迭代式开发的跨职能混合团队结构
采用迭代式开发的职能型矩阵结构
分支模型
按照职能划分分支结构
主干开发，主干发布
主干开发，分支发布
特性分支，主干发布
测试模型
独立测试部门承担功能性测试和自动化测试
在流水线中自动运行的大量的单元测试，自动化功能测试和性能测试
开发与测试的混合团队，共同承担功能性测试和自动化测试
技术架构
整体全量部署的单体应用
可独立/增量部署的单独产品
遵循SOA规范，并提供向后兼容性的微服务架构
部署模型
严格控制的发布周期和部署窗口
灰度上线，AB测试，功能开关等允许代码直接上线的实践
按需上线，开发人员直接推送代码进入生产环境
基础设施
独立运维部门严格管控的基础设施
自助化弹性扩展的Paas或IaaS平台支持的Infra as Code服务
使用脚本自动创建生产环境一致的基础设施
数据库
手工完成数据结构迁移
提供数据结构的向前/向后兼容性
差异化增量脚本完成数据迁移

6.3敏捷开发过程中的软件架构设计



敏捷软件架构的一般设计过程

6.3敏捷开发过程中的软件架构设计

- 需求分析
  - 敏捷开发中的需求分析引入了架构设计的理念，分为初始阶段需求分析和迭代阶段需求分析。
  - 初始阶段的需求分析中摒弃了具体的细节，仅仅抓住软件最高层的概念。
  - 迭代阶段需求分析是随着项目的进展逐步完善的，具有高适应性。

6.3敏捷开发过程中的软件架构设计

- 初始设计
  - 初始阶段的目标是在所有涉众之间达成关于项目的生命周期目标的协议，在项目进行之前确定重要的业务和需求风险。
  - 初始设计需要对软件系统的设计进行全局抽象层次上的考虑。
  - 包括系统的基本处理流程、系统的组织结构、模块划分、功能分配等

6.3敏捷开发过程中的软件架构设计

- 迭代过程
  - 针对需求的不可预见性，在敏捷开发中使用了迭代过程。
  - 长期的计划通常是不稳定的，单次迭代的短期计划是稳定的。
  - 迭代开发都是基于上次迭代的结果，每次迭代都有一个坚实的基础。

6.3敏捷开发过程中的软件架构设计

- 迭代过程
  - 迭代设计：是根据当前迭代过程需要完成的工作任务来进行需求分析、设计和编码等。
  - 重构：迭代过程中的重构往往发生在编码阶段，重构是对软件架构的持续改进。
  - 确定架构：迭代过程中生产出的软件（或组件）经过测试后确实能达到预期的要求，生产出了可交付的软件产品。
  - 客户交流：根据交付的可用软件，与客户进行充分交流。通过客户反馈的信息，快速有效地适应变化，在后续的迭代过程中完成客户的新要求。

6.3.4 敏捷的设计思想

- 敏捷的思想在软件架构设计中最主要的体现就是团队设计和简单设计这两种设计理念。

6.3.4 敏捷的设计思想

- 团队设计
  - 团队设计的理论依据是群体决策，这样可以避免理论上完美，但程序员无法实现的架构设计。
  - 方式：
    - 全体人员参与架构设计
    - 组织优秀的开发人员组成设计组
  - 这样设计出来的架构称为原始架构，在后续的迭代过程中不断地反馈和改进

6.3.4 敏捷的设计思想

- 团队设计
  - 优点：
    - 其结论要比个人决策更加完整，避免个人遗漏，相对稳定、周密。
  - 缺点：
    - 需要额外付出沟通成本、决策效率低、责任不明确等。

6.3.4 敏捷的设计思想

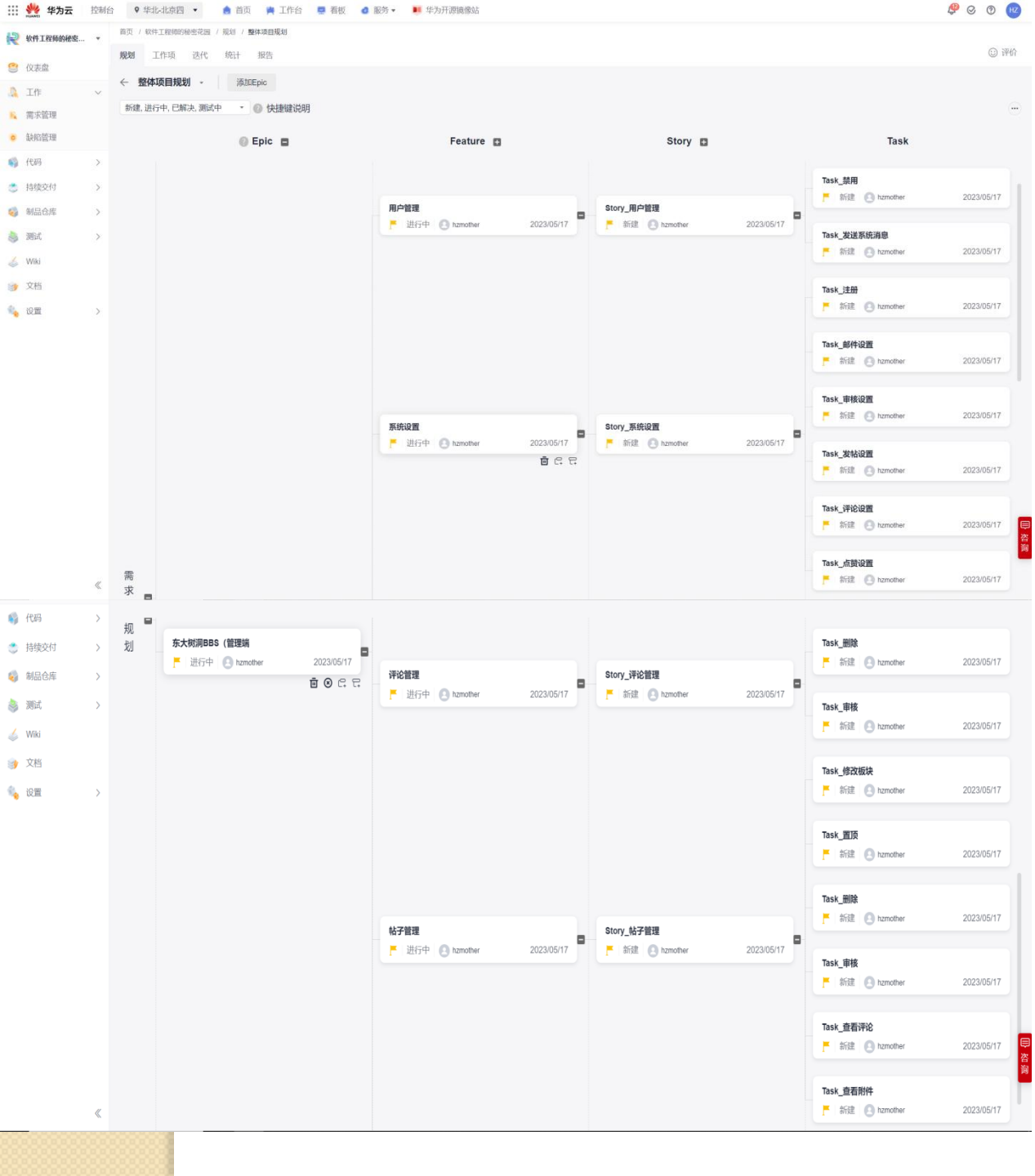
- 简单设计
  - 敏捷的思想要求软件架构设计必须是简单设计。
  - 这里的简单体现在两个方面：表达方式的简单化和现实抽象的简单化。
    - 表达方式的简单化：指的是敏捷开发中对详细架构描述文档等中间产物的弱化，只满足有效沟通即可。
    - 现实抽象的简单化：指的是仅针对当前需求建模分析，不做“多余的”工作。
  - 简单设计可以降低开发成本、提升沟通效率、增强适应性和稳定性。

6.4 两类常见敏捷软件架构设计方法

- 优秀的敏捷软件架构的设计过程一般同时包含规划式设计和演进式设计，具体体现为初始阶段设计和迭代过程中的设计。

6.4.1敏捷开发初始阶段设计

- 各种敏捷开发方法在实际应用中基本上都会在正式编码前有一个初步的设计。
- 不同方法的初始阶段设计大同小异，都是为了得到一个原始架构。
- 初始阶段设计的输出形式不同：
  - XP初始阶段输出的原始架构是以系统隐喻的方式存在的；
  - Scrum初始阶段输出的原始架构是以产品功能列表的方式存在的；
  - FDD（特征驱动的软件开发）初始阶段输出的原始架构是一个特征表。

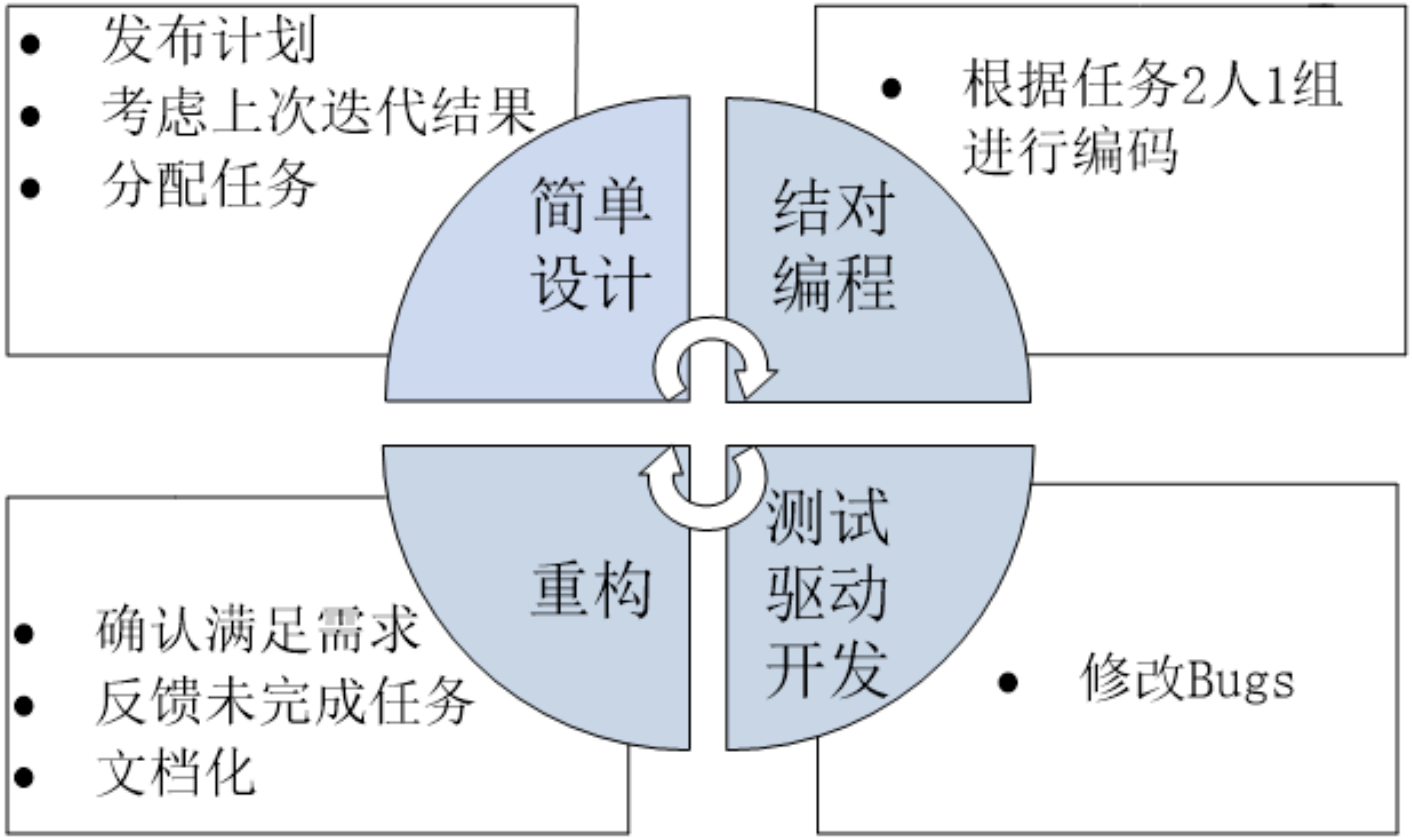

华为云
控制台
华北-北京四
首页
工作台
看板
服务
华为开源镜像站
软件工程师的秘密...
仪表盘
工作
需求管理
缺陷管理
代码
持续交付
制品仓库
测试
Wiki
文档
设置
首页 / 软件工程师的秘密花园 / 规划 / 整体项目规划
规划
工作项
迭代
统计
报告
评价
整体项目规划
添加Epic
新建, 进行中, 已解决, 测试中
快捷键说明
Epic
Feature
Story
Task
用户管理
进行中 hzmother 2023/05/17
Story_用户管理
新建 hzmother 2023/05/17
Task_禁用
新建 hzmother 2023/05/17
Task_发送系统消息
新建 hzmother 2023/05/17
Task_注册
新建 hzmother 2023/05/17
Task_邮件设置
新建 hzmother 2023/05/17
Task_审核设置
新建 hzmother 2023/05/17
系统设置
进行中 hzmother 2023/05/17
Story_系统设置
新建 hzmother 2023/05/17
Task_发帖设置
新建 hzmother 2023/05/17
Task_评论设置
新建 hzmother 2023/05/17
Task_点赞设置
新建 hzmother 2023/05/17
需求
规划
东大树洞BBS（管理端
进行中 hzmother 2023/05/17
评论管理
进行中 hzmother 2023/05/17
Story_评论管理
新建 hzmother 2023/05/17
Task_删除
新建 hzmother 2023/05/17
Task_审核
新建 hzmother 2023/05/17
Task_修改板块
新建 hzmother 2023/05/17
Task_置顶
新建 hzmother 2023/05/17
帖子管理
进行中 hzmother 2023/05/17
Story_帖子管理
新建 hzmother 2023/05/17
Task_删除
新建 hzmother 2023/05/17
Task_审核
新建 hzmother 2023/05/17
Task_查看评论
新建 hzmother 2023/05/17
Task_查看附件
新建 hzmother 2023/05/17
咨询

6.4.2 敏捷开发迭代过程中的设计

- 初始阶段输出了一个软件系统的原始架构，然后通过迭代过程进行完善。
- 迭代将致力于重用、修改、增强目前的架构，以使架构越来越强壮。

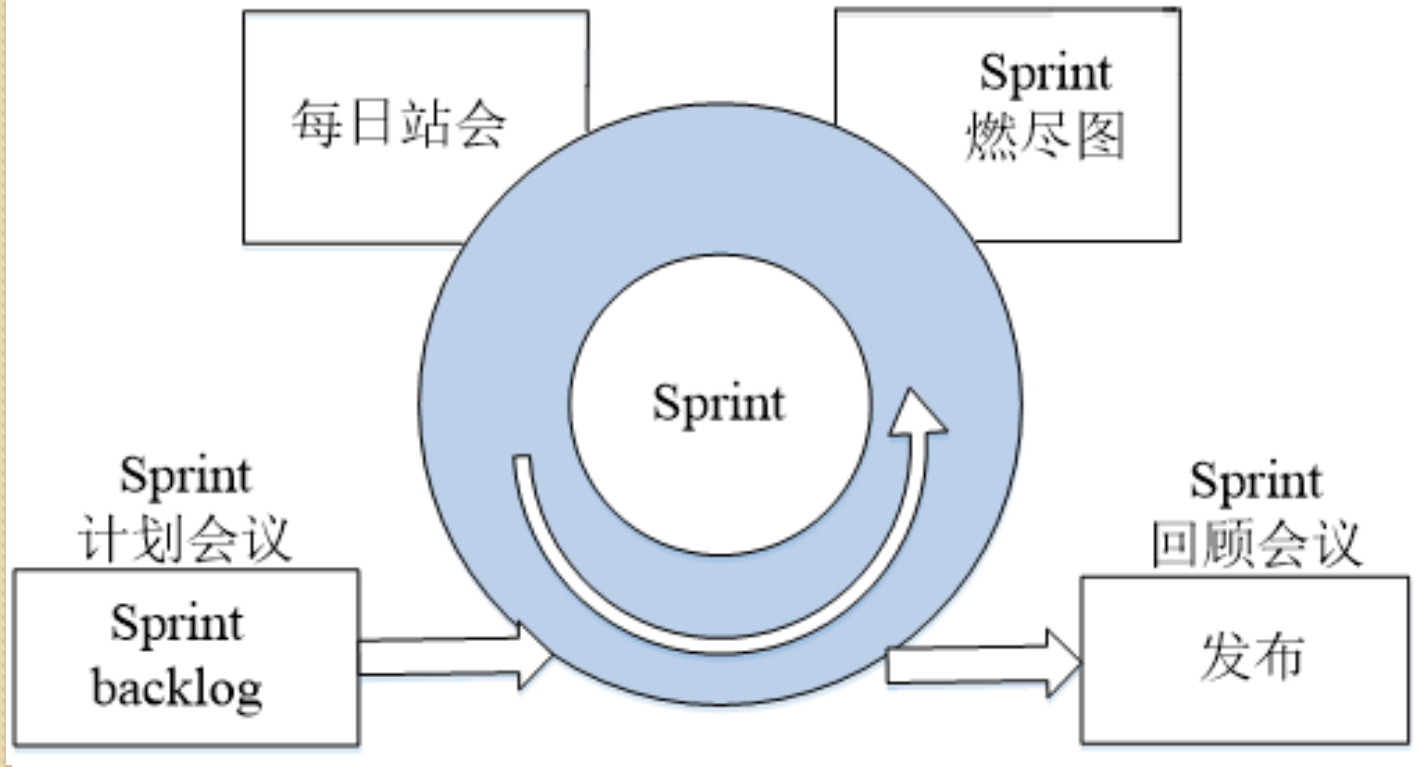
6.4.2 敏捷开发迭代过程中的设计

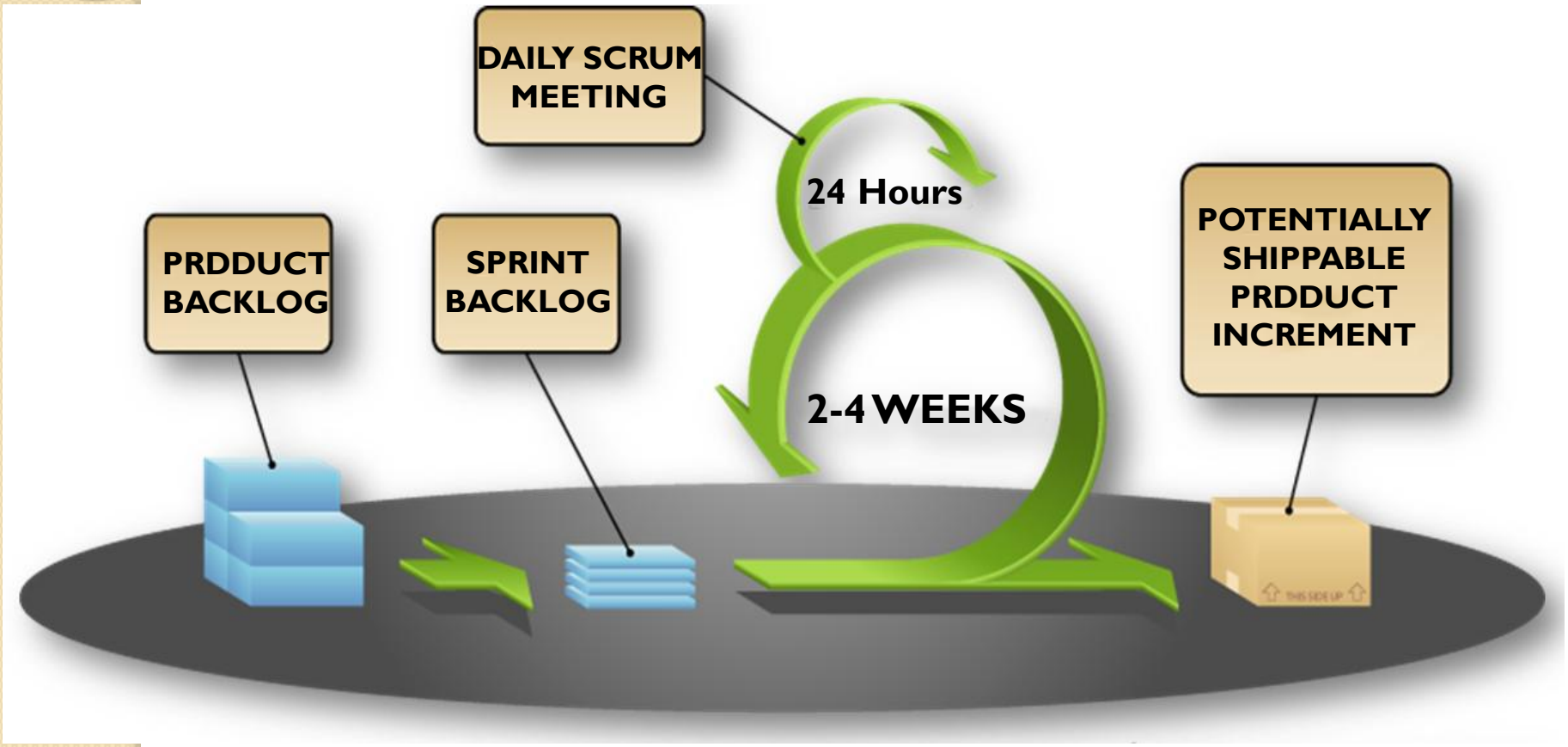
- 不同的敏捷开发方法的迭代过程不同。
- XP中的迭代过程



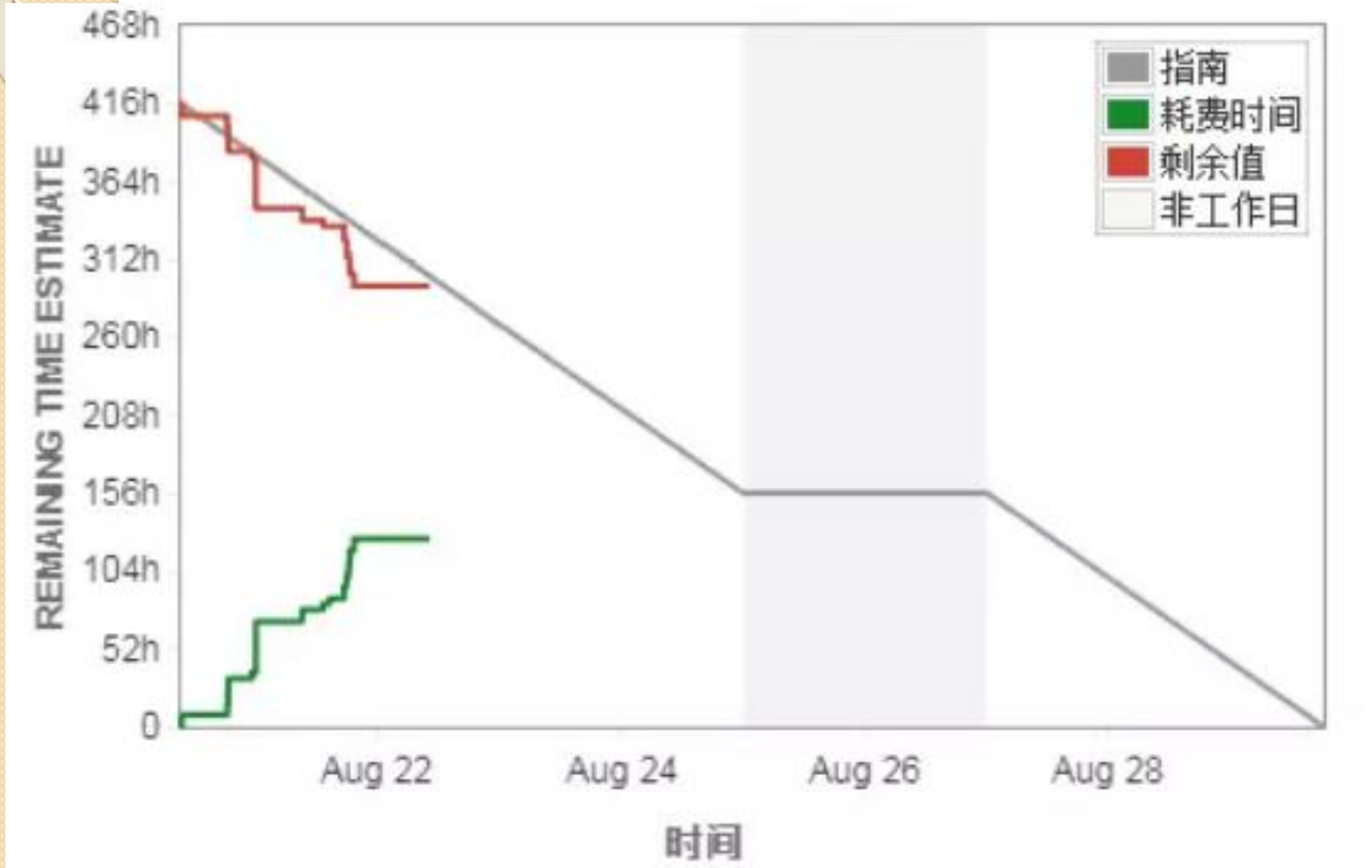
6.4.2 敏捷开发迭代过程中的设计

- 不同的敏捷开发方法的迭代过程不同
- Scrum中的迭代过程



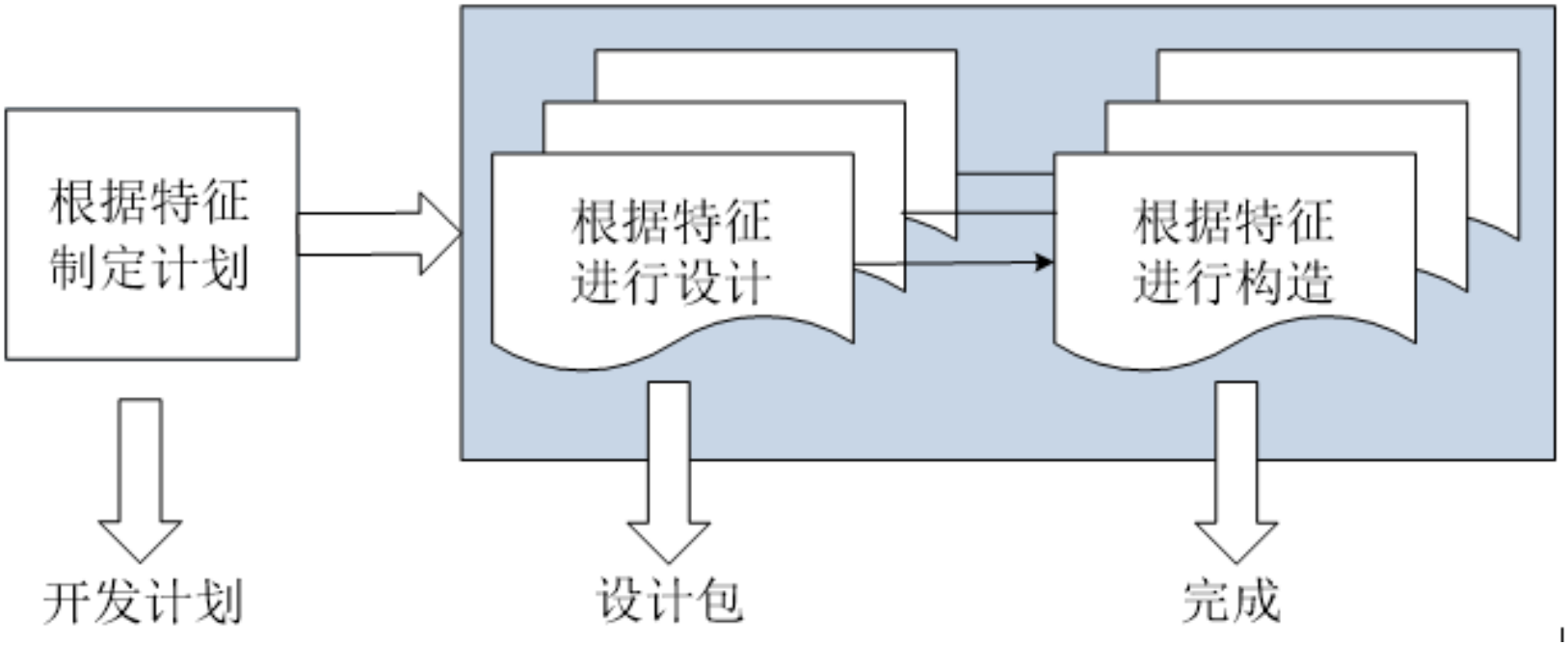

DAILY SCRUM MEETING
24 Hours
PRDDUCT BACKLOG
SPRINT BACKLOG
POTENTIALLY SHIPPABLE PRDDUCT INCREMENT
2-4 WEEKS

燃尽图


468h
416h
364h
312h
260h
208h
156h
104h
52h
0
REMAINING TIME ESTIMATE
Aug 22
Aug 24
Aug 26
Aug 28
时间
指南
耗费时间
剩余值
非工作日

6.4.2 敏捷开发迭代过程中的设计

- 不同的敏捷开发方法的迭代过程不同
- FDD中的迭代过程



6.4.2 敏捷开发迭代过程中的设计

- 三种敏捷开发方法中的软件架构特点对比

	初始阶段		迭代阶段		
	架构设计人员	架构表现方式	架构设计人员	架构表现方式	迭代设计周期
极限编程 XP	全体开发人员	系统隐喻	全体开发人员（以结对编程的2人为单位）	用户故事（隐喻）	1-3周
Scrum	Scrum Master	产品backlog	Scrum Master和全体开发人员	Springt backlog	2-4周
特征驱动软件开发 FDD	主设计师	特征表	主程序员	设计包	2周

- 如何做好软件开发初期的设计和架构? - 言理的回答 - 知乎 https://www.zhihu.com/question/24425143/answer/2648300216

6.5 本章小结

- 架构设计与针对系统做出的关键决策有关，是项目相关人员对系统内部结构和开发方式达成的共同认识。

- 软件的架构设计是没有确定答案的，面对复杂的业务场景，丰富的框架方案，架构师需要凭经验和直觉进行选择。

6.5 本章小结

- 一般说来，架构师要根据企业当前人力条件和业务约束进行对资源的合理整合，进行架构设计。
- 一个优秀成熟的架构应该有优秀的适应性并支持敏捷的思想，及时地适应需求变化并对架构做出适当的调整。
- 一个优秀的架构，其价值也许并不会直接体现在商业价值上，但它可以减少实现商业价值所需的成本。

Thanks